Doc Code: AP.PRE.REQ

PRE-APPEAL BRIEF REQUEST FOR REVIEW

Docket Number (Optional)
SVL920020078US1

I hereby certify that this correspondence is being transmitted via the USPTO EFS Web system

on

_____ January 19, 2007 _____

Signature _____ /David W. Victor/ _____

Typed or printed name _____ David W. Victor _____

Application Number	Filed
10/669,934	September 23, 2003

First Named Inventor
A. NOURI et al.

Art Unit	Examiner
2167	Miranda Le

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal.

The review is requested for the reason(s) stated on the attached five (5) sheet(s).
        Note: No more than five (5) pages may be provided.

I am the

[ ] applicant/inventor.

[ ] assignee of record of the entire interest.
See 37 CFR 3.71. Statement under 37 CFR 3.73(b) is enclosed.
(Form PTO/SB/96)

[X] attorney or agent of record.
Registration number _____ 39,867 _____

[ ] attorney or agent acting under 37 CFR 1.34.

Registration number if acting under 37 CFR 1.34 _____

_____ /David W. Victor/ _____
Signature

David W. Victor
Typed or printed name

310-553-7977
Telephone number

January 19, 2007
Date

NOTE: Signatures of all the inventors or assignees of record of the entire interest or their representative(s) are required.
Submit multiple forms if more than one signature is required, see below*.

[ ] *Total of _____ forms are submitted.

Applicant(s):	A. NOURI et al.	Examiner	Miranda Le
Serial No.	10/669,934	Group Art Unit	2167
Filed	September 23, 2003	Docket No.	SVL920020078US1
TITLE	OBJECT ORIENTED QUERY PATH EXPRESSION TO RELATIONAL OUTER JOIN TRANSLATOR METHOD, SYSTEM, ARTICLE OF MANUFACTURE, AND COMPUTER PROGRAM PRODUCT		

CERTIFICATE UNDER 37 CFR 1.8:
I hereby certify that this correspondence is being transmitted via the EFS-Web system to the U.S. Patent and Trademark Office on January 19, 2007.

__/David Victor/_____
David W. Victor

PRE-APPEAL BRIEF REQUEST FOR REVIEW ARGUMENTS

1.   Review of Rejection of Claims 1-3, 5-8, 10-13, and 15

Applicants request review of the Examiner's rejection of claims 1-3, 5-8, 10-13, and 15 as obvious (35 U.S.C. §103(a)) over Boyer (U.S. Patent No. 5,774,692) in view of Cheng (U.S. Patent No. 5,963,933).

With respect to independent claims 1, 6, and 11, Applicants submit that the cited Boyer does not teach or suggest the preamble requirement of translating a path expression in an object oriented query to a relational database outer join, said path expression comprising a navigation path through a relationship in a schema. This requirement is also part of the claim limitations, such as the last limitation, which require translating the object oriented query to a relational query. The Examiner mentioned that Boyer teaches this claim requirement but did not cite any part of Boyer as teaching or suggesting translating a path expression in an object oriented query to a relational database outer join.

Boyer mentions that the "invention provides for a new type of quantifier that is useful for object oriented queries." Boyer mentions that the "function and semantics provided by outer quantifiers is similar to that provided partly by left outer joins in relational systems" (Boyer, col. 6, lines 20-32). Although Boyer provides a quantifier that provides functions for translation similar to a left outer join in a relational system, the Examiner has not cited any part of Boyer (or the other cited Cheng) that teaches or suggests a technique for translating an object oriented query to a relational query having an outer join.

Moreover, Boyer teaches away from translating an object oriented query to an "outer join" as claimed because Boyer mentions in its "Summary of the Invention" that it is an object of the invention "to provide a service having the important aspects of the functionality of a left

outer join without incurring the difficulty of providing a left outer join implementation." (Boyer, col. 6, lines 15-29).

Applicants further request review of the Examiner's finding that col. 9, line 46 to col. 10, line 29 of Boyer teaches the claim requirement of identifying each path expression in the object oriented query which can be a candidate for a translation to an outer join. (Final Office Action, pg. 3)

The cited cols. 9-10 discuss an "outer" qualification in the object oriented query and the effect of that qualification on controlling the parser. When the "outer" keyword is recognized in the query string, the parser converts it to an internal format step, and when building the internal structure of the query, the entity created for the quantifier is marked as outer. Although the cited Boyer discusses "outer" processing for certain expressions, the Examiner has not cited any part of Boyer that teaches or suggests a path expression which is a candidate for translation to an outer join.

Applicants further request review of the Examiner finding that col. 9, line 34 and 46-65 and col. 10, line 53 to col. 11, line 54 of Boyer teach or suggest the claim requirement of ordering the path expression starting with path expression defined in a FROM clause, adding to the FROM clause path expression, each path expression identified as a candidate for a translation to an outer join, and making the ordered path expressions as input to a select operator for each level of the object oriented query. (Final Office Action, pg. 3)

The cited col. 9 mentions a FROM statement "from Supplier s, (s.parts) p", where quantifier "s" is declared on suppliers and quantifier "p" is declared on the parts that "s" supplies. A "quantifier" allows one to define an alias for a table. (Boyer, col. 1, lines 62-67) The cited col. 9 further mentions that the query requests suppliers and the parts that they supply. If a supplier supplies no parts, the supplier will not appear in the result. The cited col. 10 mentions that outer quantifiers can be implemented by a nested loop. Each quantifier in a query is implemented using an iterator or cursor that loops through a set of elements. Qualifiers over nested collections introduce a partial ordering among quantifiers.

Nowhere do the cited cols. 9 and 10 anywhere teach or suggest ordering path expressions starting with a path expression defined in a FROM clause and adding to the FROM clause path expressions, where each path expression identified for a translation to an outer join and making the ordered path expressions as input to a select operator. Instead, the cited cols. 9 and 10

discuss how qualifiers such as "outer" may apply to quantifiers in the statement. Moreover, as discussed, the cited Boyer teaches away from translating path expressions to an outer join as claimed. Consequentially, Boyer teaches away from the claim requirement of adding to a FROM statement path expressions identified as a candidate for a translation to an outer join.

The cited col. 11 discusses how if the collection of elements in "s.parts" is empty, no result is produced for that binding of "s". The plan that is generated for a query with outer quantifiers includes declarations to produce a singleton set if the set is bound to an empty set. Nowhere does this cited col. 11 anywhere teach or suggest adding to the FROM statement path expressions identified as a candidate for a translation to an outer join. Instead, the cited col. 11 discusses how the "outer" quantifier is marked on the "for loops".

Applicants request review of the Examiner's finding that the above discussed cols. 9-11 teach the claim requirement of grouping the ordered path expressions sequentially based upon a source-target dependency between ordered path expressions and based upon the identifications as a candidate for a translation to an outer join. (Final Office Action, pgs. 4, 6)

The cited col. 9 mentions the use of an "outer" qualifier that applies to quantifier "s" and elements dependent on supplier "s". Although the cited col. 9 mentions the use of dependency to determine the quantifiers to which the qualifier "outer" applies, the Examiner has not identified any part of col. 9 that teaches or suggests grouping ordered path expressions sequentially based upon a source-target dependency between ordered path expressions and upon the identification as a candidate for an outer join. Instead, the cited col. 9 looks at dependency to determine which quantifiers the "outer" qualify.

The cited cols. 10-11 discuss how to implement outer quantifiers using nested loops to mark the "for loops" with outer. The Examiner has not cited any part of cols. 10-11 that teach or mention grouping ordered path expressions sequentially based upon a source-target dependency between ordered path expressions and upon the identification as a candidate for an outer join.

Applicants further request review of the Examiner's finding that col. 10, lines 15-42 of Cheng teach or suggest the claim requirements of creating a quantifier for each path expression, said quantifier comprising a variable representing a table in a relational database and replacing each grouped path expression with a corresponding quantifier and related table in a relational database. (Final Office Action, pg. 3)

The cited col. 10 mentions transforming a full join query into a union query of left outer join and right order joint. A column of constant 1 is added in the null producing operand of the right outer join to assure that the column value is null for the preserved tuples. By applying the "IS NULL" predicate after the right outer join, all matched rows are removed from the answer set. Nowhere does this cited col. 10 anywhere teach or suggest creating a quantifier for each path expression and replacing each grouped path expression with a corresponding quantifier. Instead, the cited col. 10 discusses how to add a constant to an operand of the right outer join.

Moreover, the claims require that the quantifier is created for a path-expression in an object oriented query. Applicants submit that Cheng teaches away from translating an object oriented query to a relational query as claimed because, according to Cheng's "Field of the Invention", "[t]his invention relates to query mechanisms for a relational database management system" (Cheng, col. 1, lines 6-10). The Examiner has not cited any part of Cheng that teaches the claim requirement for creating a quantifier for each path expression as part of translating path expressions in an object oriented query.

Applicants further submit that the cited combination of Boyer and Cheng also fail to teach the last two limitations of replacing grouped path expressions with a corresponding quantifier and related table in a relational database because neither of these references alone or in combination teach or suggest a process for translating an object oriented query to a relational database query with an outer join. Thus, these references would not teach these additional claimed operations as part of such a translation.

Applicants request review of the rejection of dependent claims 2, 3, 5, 7, 8, 10, 12, 13, and 15. The dependent claims provide additional grounds of patentability over the cited Cheng and Boyer because they provide further operations for translating an object oriented query to a relational query with outer joins and inner joins. As discussed, the cited Cheng and Boyer nowhere teach or suggest operations for translating an object oriented query to a relational query with joins.

2.   Review of Rejection of Claims 4, 9, and 14

Applicants further request review of the rejection of claims 4, 9, and 14 as obvious (35 U.S.C. §103(a)) over Boyer in view of Pirahesh (U.S. Patent No. 5,548,754).

Specifically, Applicants request review of the Examiner's finding that col. 9, lines 1-8 of Pirahesh teaches the additional claim requirements that the optimization identifies a quantifier as a candidate for a translation to an inner join if a LIKE, IN, or BETWEEN operator exists in a WHERE clause containing a corresponding path expression. (Final Office Action, pgs. 5-6, 8-9)

The cited col. 9 discusses a query where an early-out join can be used on the subquery or inner query block even though the inner table column is referenced after the join in the subquery's select clause. Nowhere does the cited col. 9 anywhere teach, suggest or mention that that the optimization identifies a quantifier as a candidate for a translation to an inner join if a LIKE, IN, or BETWEEN operator exists in a WHERE clause containing a corresponding path expression.

Moreover, the "Field of the Invention" of Pirahesh mentions that the "invention" relates to optimization of SQL queries in a relational database. (Pirahesh, col. 1, lines 1-10) As with the other references, the Examiner has not cited any part of Pirahesh that teaches or suggests processing a query as part of translating an object oriented query to a relational query with outer joins as claimed.

Dated: January 19, 2007

By: /David Victor/

David W. Victor
Registration No. 39,867

Please direct all correspondences to:

David Victor
Konrad Raynes & Victor, LLP
315 South Beverly Drive, Ste. 210
Beverly Hills, CA 90212
Tel: 310-553-7977
Fax: 310-556-7984